2/5/1
DIALOG(R)File 351:Derwent WPI

011484432 **Image available**
WPI Acc No: 1997-462337/ **199743**
XRPX Acc No: N97-385046

Load distribution control system - transfers job suitably, based on selection of highest excess capacity computer, when computer attached with master computer receives job

Patent Assignee: FUJITSU LTD (FUIT)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 9212467	A	19970815	JP 9613522	A	19960130	199743	B

Priority Applications (No Type Date): JP 9613522 A 19960130
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 9212467	A		11	G06F-015/16	

Abstract (Basic): JP 9212467 A

The system disperses load to each computer (2) of a parallel computers (1). An excess capacity evaluation unit (22) evaluates the excess capacity, by receiving the load information on each computer. When one of the computers starts a job, the job information is received and the highest excess capacity computer is selected based on the output of the excess capacity evaluation unit.

This selected computer performs the received job. When the computer attached with the master computer receives the job, based on the selection of the highest excess capacity computer, the job is suitably transferred corresponding to the output of the excess capacity evaluation unit.

ADVANTAGE - Improves efficiency of processing. Enables dynamic selection of highest excess capacity computer.

Dwg.1/9

Title Terms: LOAD; DISTRIBUTE; CONTROL; SYSTEM; TRANSFER; JOB; SUIT; BASED; SELECT; HIGH; EXCESS; CAPACITY; COMPUTER; COMPUTER; ATTACH; MASTER; COMPUTER; RECEIVE; JOB
Derwent Class: T01
International Patent Class (Main): G06F-015/16
International Patent Class (Additional): G06F-009/46
File Segment: EPI

ref 4

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平9－212467

(43)公開日　平成９年(1997)８月15日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　15/16	３７０		Ｇ０６Ｆ　15/16	３７０Ｎ	
9/46	３４０		9/46	３４０Ｄ	
	３５０			３５０	

審査請求　未請求　請求項の数３　ＯＬ　（全 11 頁）

(21)出願番号　特願平8－13522

(22)出願日　平成８年(1996)１月30日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　伊藤　雅典
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　岡田　守弘

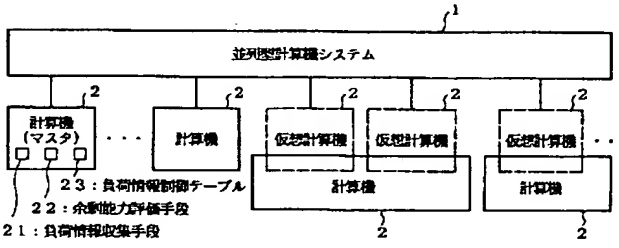
(54)【発明の名称】　負荷分散制御システム

(57)【要約】

【課題】　本発明は、負荷分散制御システムに関し、計算機ごとにＣＰＵ能力および台数が異なりしかも仮想計算機が混在する並列型計算機システムであってもバッチジョブを実行する計算機を自動的に選択してジョブを配送し負荷分散を図り、並列型計算機システムの処理効率を高めることを目的とする。

【解決手段】　各計算機の負荷情報の通知を受けて余剰能力を評価して算出する余剰能力評価手段と、いずれかの計算機にジョブが投入されたときに当該ジョブ情報の通知を受けて評価した余剰能力が最も高い計算機を選択し、ジョブを受け付けた計算機が最も余剰能力が高いときはその計算機にジョブを実行させ、一方、ジョブを受け付けた計算機以外の他の計算機が最も余剰能力が高いときはそのジョブを受け付けた計算機にジョブを転送させて実行させる手段とを並列型計算機のうちのマスタ計算機に備えるように構成する。

本発明のシステム構成図



1

【特許請求の範囲】

【請求項１】並列型計算機の各計算機に負荷を分散する負荷分散制御システムにおいて、

各計算機（仮想計算機を含む、以下同様）の負荷情報の通知を受けて余剰能力を評価して算出する余剰能力評価手段と、

いずれかの計算機にジョブが投入されたときに当該ジョブ情報の通知を受けて上記評価した余剰能力が最も高い計算機を選択し、ジョブを受け付けた計算機が最も余剰能力が高いときはその計算機にジョブを実行させ、一方、ジョブを受け付けた計算機以外の他の計算機が最も余剰能力が高いときはそのジョブを受け付けた計算機にジョブを転送させて実行させる手段とを並列型計算機のうちのマスタ計算機に備えたことを特徴とする負荷分散制御システム。

【請求項２】実記憶に対する負荷が所定負荷よりも高くて過負荷と判明したときにその計算機を除外して他の計算機について余剰能力を評価して算出することを特徴とする請求項１記載の負荷分散制御システム。

【請求項３】上記余剰能力として、各計算機のＣＰＵ処理時間＋ＣＰＵ待ち時間、あるいはＣＰＵ処理時間＋ＣＰＵ待ち時間＋Ｉ／Ｏ処理時間＋Ｉ／Ｏ待ち時間としたことを特徴とする請求項１あるいは請求項２記載の負荷分散制御システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、並列型計算機の各計算機に負荷を分散する負荷分散制御システムに関するものである。

【０００２】近年の計算機システムにおいて、業務処理容量の増大、処理速度の高速化、連続運用の必要性、信頼性の向上が要求されている。このため、仮想計算機を混在でき、システムの運用中に動的に稼働計算機の数を増減できる並列型計算機システムが提供されている。この並列型計算機システムでは、ＣＰＵ能力の異なる計算機とＣＰＵ能力が可変の計算機が混在し、計算機によってＣＰＵ負荷状態が異なり、かつ並列計算機システムで稼働する計算機の数が動的に変動するため、並列型計算機システム全体の処理容量と処理速度を高めるためには各計算機のＣＰＵ負荷を平準化する必要がある。しかし、ユーザが各計算機のＣＰＵ負荷を平準化するのは困難であるために、システムが自動的にバッチジョブを実行する計算機を決定することが望まれている。

【０００３】

【従来の技術】従来の並列型計算機システムは、図９に示すように、同じ能力のＣＰＵを同じ台数備えた計算機を並列に接続する。この並列型計算機システムにおける負荷分散を行う手法は、ＯＳが自計算機のＣＰＵ使用率を求めて他の全ての計算機に送信して知らせる。そして、ジョブがある計算機に投入された場合、最もＣＰＵ使用率の低い計算機が余剰ＣＰＵ能力が大きいとみなしてジョブを配送し、その計算機にジョブの実行をさせる（自計算機の余剰ＣＰＵ能力が大きいときは自計算機がジョブを実行する）。

2

【０００４】

【発明が解決しようとする課題】上述した並列型計算機システムでは、並列型計算機システムを構成する計算機ごとに１台あたりのＣＰＵ能力が違う場合や、ＣＰＵごとの能力が同じでも台数が違う場合には、ＣＰＵ使用率が同じでもあっても余剰ＣＰＵ能力が同じとは限らないから、ＣＰＵ使用率の最も低い計算機にジョブを配送しても、ジョブを最も高速に実行できるとはならない問題が発生する。

【０００５】また、並列型計算機システムには仮想計算機が混在できるが、仮想計算機には通常、次の３種類の動作モードがある。

・ＡＵＴＯモード：同じ計算機上で走行する他の仮想計算機が要求するＣＰＵ能力と競合しない限り必要なだけ可変にＣＰＵ能力を使用できるモードである。

【０００６】・上限ＡＵＴＯモード：ＡＵＴＯモードの仮想計算機と同じ計算機上で共存可能で、決められた比率（ＣＰＵ配分比）しかＣＰＵ能力を使用できないモードである。

【０００７】・ロジカルモード：ＡＵＴＯモードや上限ＡＵＴＯモードとは同じ計算機上で共存できないが、１台の計算機のＣＰＵ能力を任意の固定比率（ＣＰＵ配分比）に分割して使用するモードである。

【０００８】これらの仮想計算機システムはそれぞれ独立に、その上で走行するソフトウェアから認識できるＣＰＵ（論理ＣＰＵ）の数を定義することができる。このため、上限ＡＵＴＯモードとロジカルモードの仮想計算機は、実ＣＰＵの能力のうち配分比だけのＣＰＵ能力を持つ計算機と見なせるから、上記した問題が発生する。また、仮想計算機システムは、動的にＣＰＵ配分比を変更でき、上限ＡＵＴＯモードとロジカルモードの仮想計算機において、異なる時刻に同じＣＰＵ使用率であっても、余剰ＣＰＵ能力が同じとは限らないという問題も発生する。

【０００９】また、計算機の実記憶負荷が高く、仮想計算機が過剰に動作し、外部ページと実ページで過剰に交換が行われている場合には、ページングＩ／Ｏ待ちが頻発してＣＰＵ使用率が低くなることがあり、この場合に、新たにジョブを動作させると、実記憶負荷が更に高くなり、ページングＩ／Ｏ待ちがより頻繁に発生し、ＣＰＵ使用率が更に低くなって仮想計算機の処理効率が低下してしまう問題が発生する（この場合には、従来の負荷分散の手法では、ジョブを配送するのは逆効果となってしまう点で問題である）。

【００１０】本発明は、これらの問題を解決するため、計算機ごとにＣＰＵ能力および台数が異なりしかも仮想

計算機が混在する並列型計算機システムであってもバッチジョブを実行する計算機を自動的に選択してジョブを配送し負荷分散を図り、並列型計算機システムの処理効率を高めることを目的としている。

【００１１】

【課題を解決するための手段】図１を参照して課題を解決するための手段を説明する。図１において、並列型計算機システム１は、複数の計算機２および仮想計算機２を相互に接続して並列処理を行うためのものである。

【００１２】計算機２および仮想計算機２は、ジョブを実行するものである。仮想計算機２は、計算機（実計算機）２に任意の個数を動的に設けることができる。ここで、計算機（マスタ）２は、負荷情報収集手段２１、余剰能力評価手段２２などから構成されるものである。

【００１３】負荷情報収集手段２１は、計算機（スレーブ）２から負荷情報などを収集するものである。余剰能力評価手段２２は、各計算機から収集した負荷情報をもとに当該計算機の余剰能力を評価して算出するものである。

【００１４】次に、動作を説明する。計算機（マスタ）２の負荷情報収集手段２１が各計算機から負荷情報を収集し、余剰能力評価手段２２がこの収集した負荷情報をもとに余剰能力を評価して算出し、いずれかの計算機２にジョブが投入されたときに当該ジョブ情報の通知を受けた計算機（マスタ）２が評価した余剰能力が最も高い計算機２を選択し、ジョブを受け付けた計算機２が最も余剰能力が高いときはその計算機２にジョブを実行させ、一方、ジョブを受け付けた計算機以外の他の計算機２が最も余剰能力の高いときはそのジョブを受け付けた計算機２にジョブを転送させ実行させるようにしている。

【００１５】この際、実記憶に対する負荷が所定負荷よりも高くて過負荷と判明したときにその計算機２を除外して他の計算機２について余剰能力を評価して算出し、最も余剰能力の高い計算機にジョブを実行させるようにしている。

【００１６】また、余剰能力として、各計算機２のＣＰＵ処理時間＋ＣＰＵ待ち時間、あるいはＣＰＵ処理時間＋ＣＰＵ待ち時間＋Ｉ／Ｏ処理時間＋Ｉ／Ｏ待ち時間として算出するようにしている。

【００１７】従って、計算機ごとにＣＰＵ能力および台数が異なり、しかも仮想計算機が混在する並列型計算機システムであってもバッチジョブを実行する計算機（仮想計算機を含む）２を自動的に選択してジョブを配送し負荷分散を図ることにより、並列型計算機システムの処理効率を高めることが可能となる。

【００１８】

【発明の実施の形態】次に、図１から図８を用いて本発明の実施の形態および動作を順次詳細に説明する。

【００１９】図１は、本発明のシステム構成図を示す。

図１において、負荷情報制御テーブル２３は、計算機２から収集した負荷情報を記憶したり、これら記憶した負荷情報からジョブを実行させる計算機２を選択する時点で余剰能力としてエラップス期待値を計算して設定するものである（後述する図５参照）。ここで、エラップス期待値の最も小さい計算機２にジョブを配送するようにする。

【００２０】次に、図２を用いて計算機（マスタ）２および図３を用いて計算機（スレーブ）２の構成を順次詳細に説明する。図２は、本発明の計算機（マスタ）例を示す。これは、図１の計算機（マスタ）２の詳細構成図であって、図１の負荷情報収集手段２１は図２のＣＰＵ負荷情報収集部１０に対応し、図１の余剰能力評価手段２２は図２のジョブ配送先決定部９の一部に含まれるものである。

【００２１】図２において、ＯＳ５は、オペレーティングシステムであって、全体を統括制御するものであり、ジョブ実行部７、ジョブ配送部８、ジョブ配送先決定部９、ＣＰＵ負荷情報収集部１０、ＣＰＵ負荷情報受信部１２、ジョブ配送先通知部１３、ジョブ受付部１４、ジョブ情報受信部１６などから構成されるものである。

【００２２】ジョブ配送部８は、ジョブ配送先決定部９によって決定された配送先にジョブを配送するものである。ジョブ配送先決定部９は、余剰能力の最も高い（エラップス値の最も小さい）計算機をジョブ配送先と決定するものである。

【００２３】ＣＰＵ負荷情報収集部１０は、計算機２の負荷情報を収集するものである。ＣＰＵ負荷情報受信部１２は、スレーブ計算機２よりＣＰＵ負荷情報を受信するものである。

【００２４】ジョブ配送先通知部１３は、ジョブ配送先決定部９によって決定された配送先の計算機２にジョブを配送するようにスレーブ計算機に通知するものである。ジョブ受付部１４は、投入されたジョブを受け付けるものである。

【００２５】ジョブ情報受信部１６は、スレーブ計算機に投入されたジョブの情報を受信するものである。図３は、本発明の計算機（スレーブ）例を示す。これは、図１の計算機（マスタ）２以外のスレーブの計算機２の詳細構成図であある。

【００２６】図３において、ＯＳ５は、オペレーティングシステムであって、全体を統括制御するものであり、ジョブ実行部７、ジョブ配送部８、ＣＰＵ負荷情報収集部１０、ジョブ受付部１４、ジョブ配送先受信部１６、ジョブ情報通知部１７などから構成されるものである。７、８、１０、１４は図２と同一であるので説明を省略する。

【００２７】図３において、ＣＰＵ負荷情報通知部１１は、ＣＰＵ負荷情報などをマスタ計算機に通知するものである。ジョブ配送先受信部１５は、ジョブの配送先を

5

マスタ計算機から受信するものである。

【００２８】ジョブ情報通知部１７は、投入されたジョブのジョブ情報をマスタ計算機へ通知するものである。以下図４ないし図８を用いて図１ないし図３の構成の動作を順次詳細に説明する。

【００２９】図４は、本発明の動作説明図（その１）を示す。図４の（ａ）は、負荷情報収集のフローチャートを示す。図４の（ａ）において、ステージ１は、任意の計算機で実行するものである。

【００３０】Ｓ１は、自計算機の負荷情報を収集する。この負荷情報は、例えば図４の（ｂ）に示す　ないし　の情報を収集する。Ｓ２は、マスタ計算機に通知する。

【００３１】図４の（ａ）において、ステージ２は、マスタ計算機で実行するものである。Ｓ３は、各計算機の負荷情報を受信する。Ｓ４は、負荷情報制御テーブルに格納する。

【００３２】以上のステージ１のＳ１、Ｓ２およびステージ２のＳ３、Ｓ４によって、全ての計算機２の負荷情報およびＩ／Ｏ負荷情報を計算機（マスタ）２が収集し、後述する図５の負荷情報制御テーブル２３のように設定（Ｉ／Ｏ負荷情報は未設定）できたこととなる。

【００３３】図４の（ｂ）は、ステージ１において収集・通知する負荷情報の例を示す。負荷情報は、図示のないし　の下記のものである。

計算機識別子：並列型計算機システムを構成する各計算機を識別するもの

ＣＰＵ能力　：実ＣＰＵの能力（実ＣＰＵ一台当たりのＭＩＰＳ値）

ＣＰＵ台数（１～Ｎ）

実ＣＰＵ台数（１～Ｍ）

ＣＰＵ使用率

実ＣＰＵ使用率

＝ＣＰＵ処理時間＋ＣＰＵ待ち時間
＋Ｉ／Ｏ処理時間＋Ｉ／Ｏ待ち時間　　（式１）
＝ＣＰＵ処理時間×（１＋α（ＣＰＵ数、ＣＰＵ使用率））
＋Ｉ／Ｏ処理時間×（１＋β（チャネル数、チャネル使用率））　（式２）

αとβは待ち行列理論の一般論から導かれるものである。このαとβとの関係は、例えばＣＰＵバウンドなジョブであれば、必然的にＣＰＵ処理時間が大きくなり、Ｉ／Ｏ処理時間が小さくなるので、αの大小関係に敏感に、βの大小関係に鈍感になり、ＣＰＵ負荷情報およびＩ／Ｏ負荷情報をまとめて計算機２の余剰能力を評価してエラップス値として算出することが可能となった。

【００３６】尚、Ｉ／Ｏ処理は、チャネルと呼ばれる入出力機構を経由してディスク装置などと主記憶との間でデータの転送を行っている。１回のＩ／Ｏ処理にかかる時間は、チャネル数やチャネルの使用率に影響されるので、上記（式２）に示すようにＣＰＵの場合と同様に評価するようにしている。

【００３７】図６は、本発明の動作説明図（その２）を

6

ＣＰＵ配分比：仮想計算機への実ＣＰＵ能力の配分比（動的変更可能）

計算機構成情報：実計算機、ＡＵＴＯモード／上限ＡＵＴＯモード／ロジックモードの仮想計算機の区別を表示

実記憶負荷情報：スラッシングを起こしているか否かを表示

図４の（ｃ）は、余剰能力の評価のフローチャートを示す。

【００３４】図４の（ｃ）において、ステージ１は、任意の計算機で実行するものである。Ｓ１１は、ユーザが任意の計算機にジョブを投入する。Ｓ１２は、ジョブが投入された計算機がマスタ計算機か、スレーブ計算機か判別する。マスタ計算機の場合には、ステージ２（図４の（ａ）のステージ２）に進む。一方、スレーブ計算機の場合には、Ｓ１３でマスタ計算機に、投入されたジョブの情報を通知する。

【００３５】以上のＳ１１、Ｓ１２によって、計算機に投入されたジョブ情報が全てマスタ計算機に通知されたこととなる。図５は、本発明の負荷情報制御テーブル例を示す。この負荷情報制御テーブル２３は、既述した図４のＳ４で、全ての計算機２から収集された負荷情報を設定して記憶したものであって、既述した図４の（ｂ）の　ないし　の情報を設定して記憶したものである。図中の“エラップス期待値”は、　ないし　の情報をもとに算出したものであって、計算機の余剰能力を表すものであり、小さいほど、計算機の余剰能力が高いものである。このエラップス期待値は、例えば下記の式によって計算する（尚、図５の負荷情報制御テーブル２３は、ＣＰＵ処理時間＋ＣＰＵ待ち時間についてのものである）。ジョブ投入からジョブ終了までに必要な時間（エラップス期待値）は、

示す。図６において、ステージ２は、マスタ計算機で実行するものである。Ｓ２１は、任意の計算機より、投入されたジョブの情報を受信する。これは、スレーブ計算機に投入されたジョブの情報をマスタ計算機が受信、およびマスタ計算機に投入されたジョブの情報を受け付け、既述した図５の負荷情報制御テーブル２３に設定する。

【００３８】Ｓ２２は、負荷情報制御テーブルを参照し、各計算機の負荷情報を順々に取り出す。Ｓ２３は、実記憶負荷が過負荷か判別する。ＹＥＳの場合には、Ｓ２４ないしＳ２６をスキップしてＳ２７に進む。一方、ＮＯの場合には、Ｓ２４に進む。

【００３９】Ｓ２４は、計算機の種別を判別する。

・実計算機またはＡＵＴＯモードの仮想計算機の場合に

は、Ｓ２６に示す下記の式４によって当該計算機のエラップス期待値を評価し、Ｓ２７に進む。

（１＋α（実ＣＰＵ数、実ＣＰＵ使用率））／ＣＰＵ能力　（式４）

・上限ＡＵＴＯモード仮想計算機またはロジカルモード仮想計算機の場合には、Ｓ２５に示す下記の式５によって当該計算機のエラップス期待値を評価し、Ｓ２７に進む。

【００４１】

（１＋α（実ＣＰＵ数、実ＣＰＵ使用率））／（ＣＰＵ能力×ＣＰＵ配分比）　（式５）

Ｓ２７は、負荷情報制御テーブルの最後まで評価を行ったか判別する。ＹＥＳの場合には、Ｓ２８に進む。ＮＯの場合には、Ｓ２２に戻り、繰り返す。

【００４２】Ｓ２８は、エラップス期待値が最小かつ実記憶過負荷でない計算機をジョブ配送先として選択する。Ｓ２９は、ジョブを受け付けた計算機に、選択した計算機をジョブ配送先として通知する。そして、図７のステージ３へ進む。

【００４３】以上のＳ２１からＳ２９によって、全ての計算機から受信した負荷情報を負荷情報制御テーブル２３に設定した後、先頭から順番に取り出して実記憶負荷が過負荷でない場合に計算機の種別によって分けてそれぞれエラップス期待値を式４あるいは式５によって計算し、エラップス期待値が最も小さい計算機にジョブを配送させて転送するように通知することが可能となる。そして、後述する図７のステージ３によってジョブをエラップス期待値の最も小さい（余剰能力の最も高い）計算機に転送して実行させることが可能となる。

【００４４】図７は、本発明の動作説明図（その３）を示す。図７において、ステージ３は、ジョブを受け付けた計算機で実行するものである。

【００４５】Ｓ３１は、マスタ計算機よりジョブ配送先を受信し、ジョブをマスタ計算機から指示された計算機に配送する。ステージ４は、ジョブを配送された計算機で実行するものである。

【００４６】Ｓ４１は、ジョブを受け付けた計算機より配送されたジョブを実行する。以上のＳ３１、Ｓ４１によって、エラップス期待値の最も小さい計算機にジョブを配送して実行させることが可能となる。

【００４７】図８は、本発明の余剰能力の評価例を示す。これは、ＣＰＵ負荷情報のみをもとに余剰能力を評価して計算したものである。以下説明する。ここで、あるサービスを行う複数の窓口に対する客の到着頻度がランダム到着に従い、そのサービス量が指数分布に従うと仮定すると、到着した客がサービスを受けるまでの待ち時間は、処理時間×（窓口の数と窓口の平均稼働率の関数）として図８の（式６）として書けることが、待ち行列理論の一般論として知られている。今、窓口の稼働率

【００４０】

をＣＰＵ使用率に、窓口の数をＣＰＵ台数に対応させ、あるジョブを実行完了するために必要な時間をＣＰＵ待ち時間とＣＰＵ処理時間だけで評価すれば、ＣＰＵ処理時間は、（式７）のように評価でき、ジョブの処理に必要なダイナミックステップ数はどの計算機で走行させても同じだから除外して考えると、ジョブを投入してから終了するまでに要する時間に比例する量が得られる。この値（エラップス期待値）が最も小さく、かつ実記憶負荷が過負荷状態でない計算機に対してジョブを配送し、実行させる。ここで、Ｉ／Ｏ処理時間およびＩ／Ｏ待ち時間を既述した（式１）、（式２）のように含めるようにしてもよい。

【００４８】

【発明の効果】以上説明したように、本発明によれば、計算機ごとにＣＰＵ能力および台数が異なり、しかも仮想計算機が混在する並列型計算機システムであってもバッチジョブを実行する計算機（仮想計算機を含む）２を自動的に選択してジョブを配送し負荷分散を図る構成を採用しているため、仮想計算機を含む並列型計算機システムのバッチジョブを実行させる際に、最も余剰能力の高い計算機を動的に選択してジョブを配送し実行させ、処理効率を高めることができる。

【図面の簡単な説明】

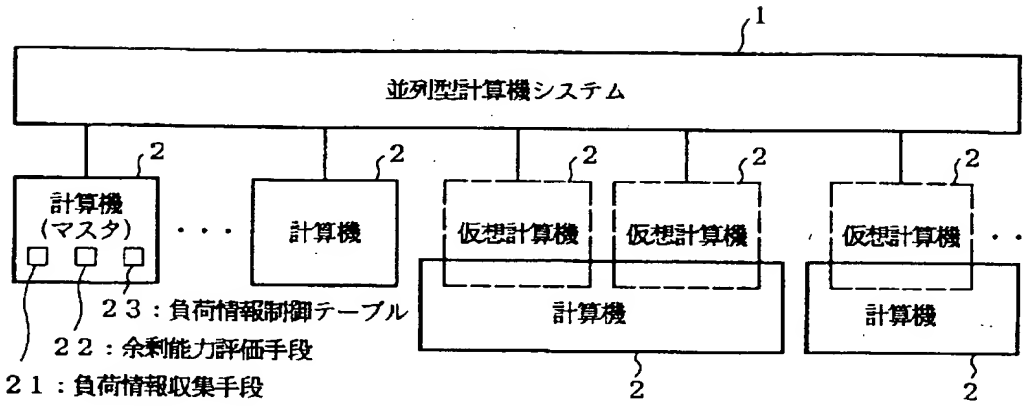
【図１】本発明のシステム構成図である。
【図２】本発明の計算機（マスタ）例である。
【図３】本発明の計算機（スレーブ）例である。
【図４】本発明の動作説明図（その１）である。
【図５】本発明の負荷制御情報テーブル例である。
【図６】本発明の動作説明図（その２）である。
【図７】本発明の動作説明図（その３）である。
【図８】本発明の余剰能力の評価例である。
【図９】従来技術の説明図である。

【符号の説明】

１：並列型計算機システム
２：計算機、仮想計算機
２１：負荷情報収集手段
２２：余剰能力評価手段
２３：負荷情報制御テーブル

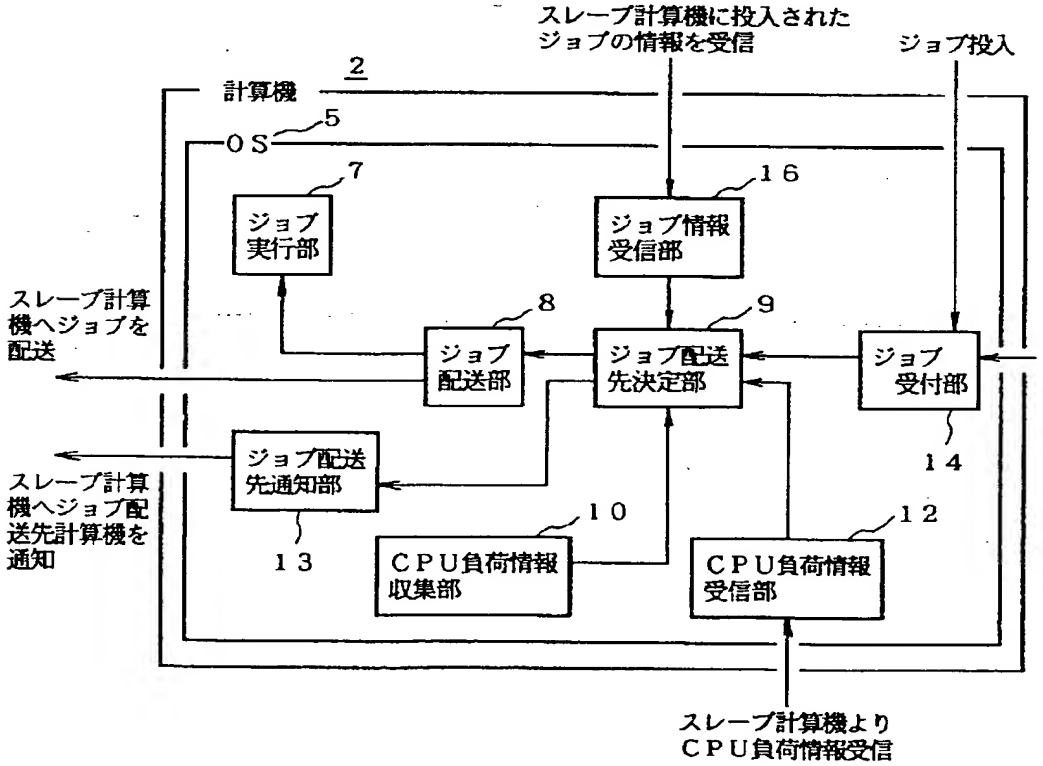
【図１】

本発明のシステム構成図



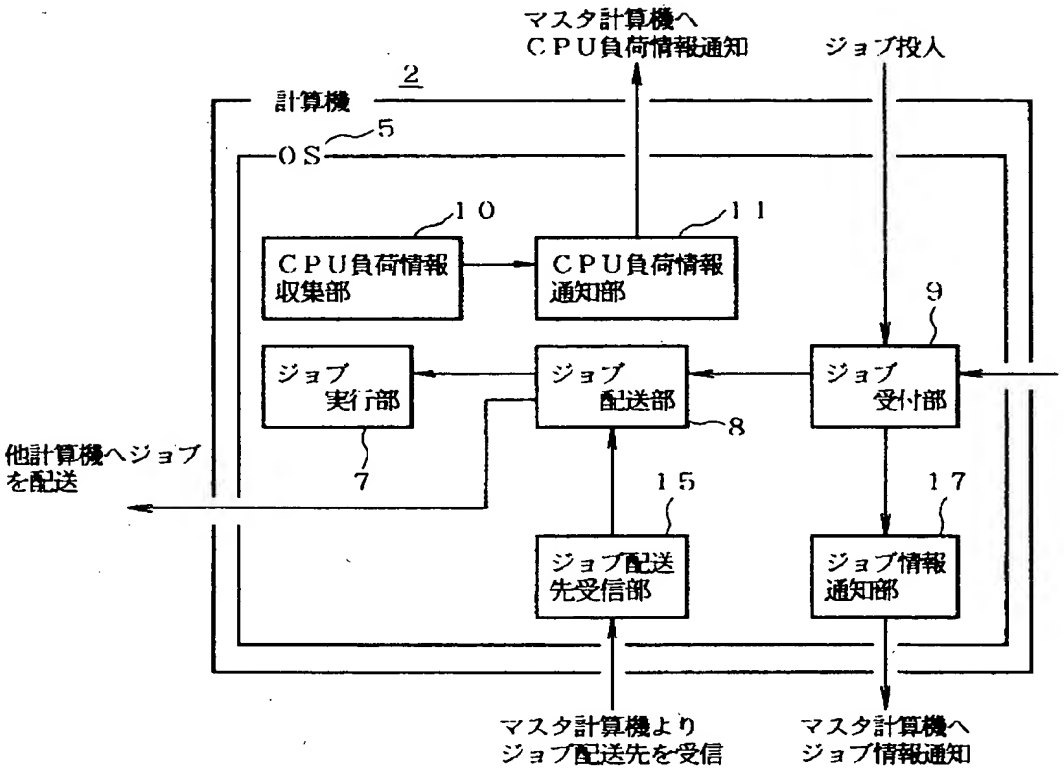
【図２】

本発明の計算機（マスタ）例



【図３】

本発明の計算機（スレーブ）例



【図５】

本発明の負荷情報制御テーブル例

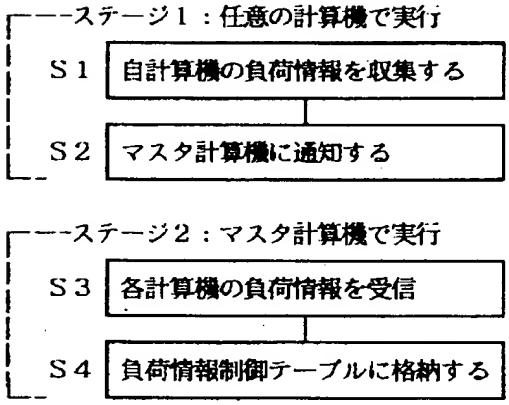
②～⑨より計算
↓
23

①計算機識別子	②	③	④	⑤	⑥	⑦	⑧計算機構成情報	⑨	エラップス期待値
１１	200	2	2	150	150	－	実計算機	OFF	１００
:	:	:	:	:	:	:	:	:	:
１ｉ	100	1	1	50	50	－	実計算機	ON	－
２１	200	2	2	120	190	100	AUTOモード仮想	OFF	２１０
２２	200	1	2	40	190	50	上限AUTOﾓｰﾄﾞ 仮想	OFF	１０
:	:	:	:	:	:	:	:	:	:
２ｊ	200	1	2	30	190	50	ﾛｼﾞｶﾙﾓｰﾄﾞ 仮想	OFF	２０
:	:	:	:	:	:	:	:	:	:

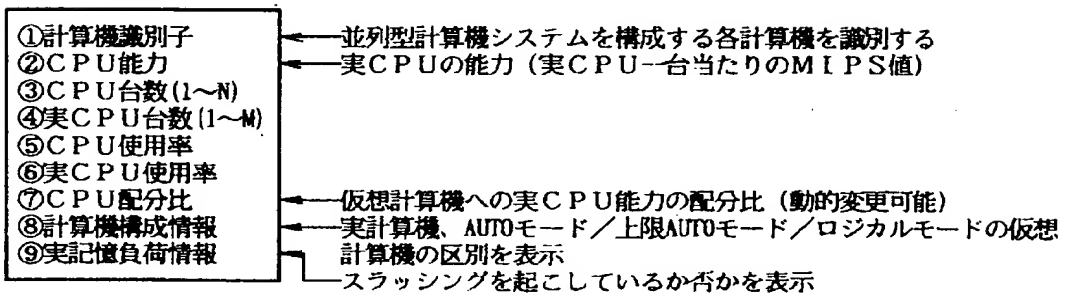
【図４】

本発明の動作説明図（その１）

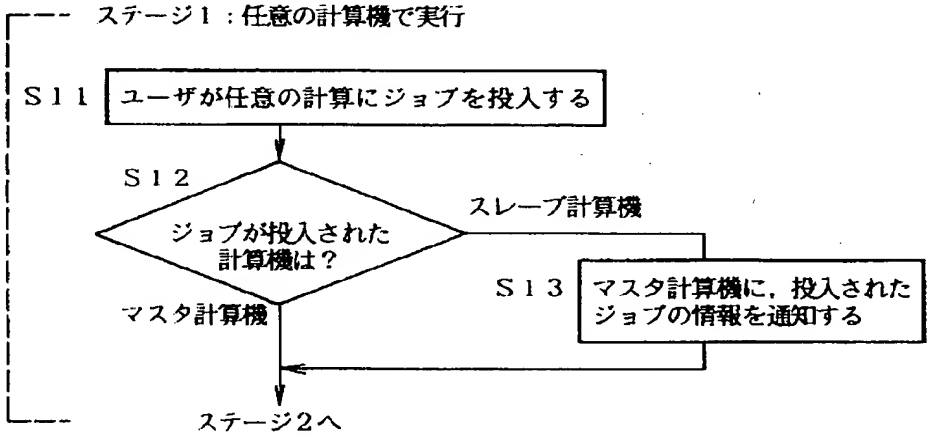
（ａ）負荷情報収集



（ｂ）ステージ１において収集・通知する負荷情報

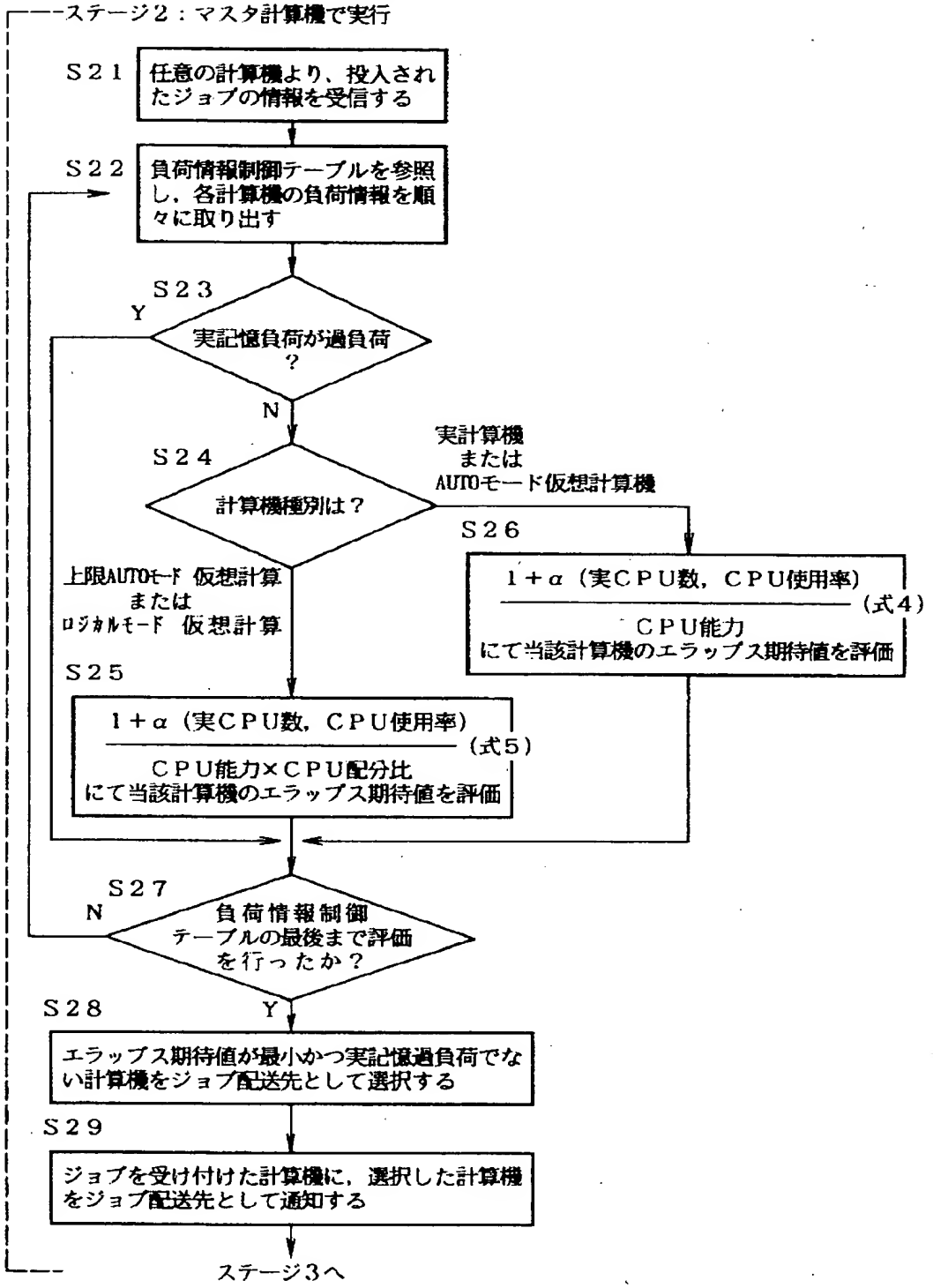


（ｃ）余剰能力評価



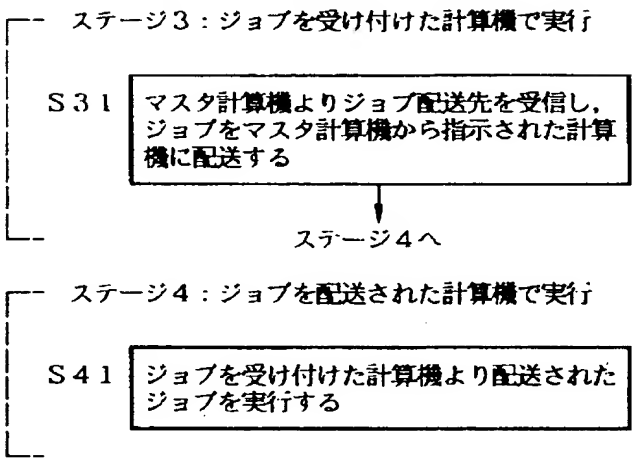
【図６】

本発明の動作説明図（その２）



【図７】

本発明の動作説明図（その３）



【図８】

本発明の余剰能力の評価例

（式６）

待ち時間　＝　処理時間　×　α（窓口の数（ｓ），窓口の平均稼働率（ａ））

$$= \text{処理時間} \times \frac{a^s}{(s-1)! \times (s-a)^2} \times \frac{1}{\sum_{n=0}^{s-1} \frac{a^n}{n!} - \frac{a^s}{(s-1)! \times (s-a)}}$$

$$0 \leq \text{窓口の平均稼働率（a）} \leq 100$$

（式７）

ＣＰＵ待ち時間＋ＣＰＵ処理時間

＝ＣＰＵ処理時間×α＋ＣＰＵ処理時間

＝ＣＰＵ処理時間×（１＋α）

ここでＣＰＵ処理時間は，

$$\text{ＣＰＵ処理時間} = \frac{\text{ジョブを処理完了するのに必要なダイナミックステップ数}}{\text{ＣＰＵ能力（ＭＩＰ値）}}$$

だから，最終的に（式４）を以下のように評価する

$$\text{ＣＰＵ処理時間＋ＣＰＵ待ち時間} \propto \frac{(1+\alpha\ (\text{ＣＰＵ台数，ＣＰＵ使用率}))}{\text{ＣＰＵ能力}}$$

【図９】

従来技術の説明図